

Documentación Técnica
Accuracy V1 – Nueva Funcionalidad
23-04-2021
Revisión: 01

La nueva funcionalidad se requiere para hacer accuracy personalizado, según como se requiera, tareas, modelos o labels específicos.

El cambio que se realizó permite tener esta facilidad para hacer el accuracy por elementos específicos y medir solo los necesarios según el reconocimiento de la foto.

Modulo Modificado.

1. ebisuu_accuracy

Archivos Modificados.

Modelos:

1. ebisuu_accuracy/models.py

Views:

1. ebisuu_accuracy/views/new_accuracy.py
2. ebisuu_accuracy/views/utils.py
3. ebisuu_accuracy/views/accuracy_photo_global.py
4. ebisuu_accuracy/views/accuracy_photo_task.py

Urls:

1. ebisuu_accuracy/urls/utils.py

Templates:

1. app/templates/app/accuracy/v1/create_new_accuracy.html
2. app/templates/app/accuracy/v1/edit_new_accuracy.html

Modificaciones específicas por archivo.

Models.py: Se agregaron 2 campos en el modelo NewAccuracy() y se agregaron funciones al modelos para traer la información de los nuevos modelos.

Nuevos campos:

Campo: category_labels.

Tipo: ManyToMany.

Relación: LabelCategory

Campo: labels_available

Tipo: ManyToMany,.

Relación: labels_available

Nuevas Funciones dentro del modelo:

- `get_category_labels()`
- `get_name_category_labels()`
- `get_ids_labels()`
- `get_name_labels()`,

```
def get_category_labels(self):  
    return [a.pk for a in self.category_labels.all()]  
  
def get_name_category_labels(self):  
    return ", ".join([a.name for a in self.category_labels.all()])  
  
def get_ids_labels(self):  
    return [a.pk for a in self.labels_available.all()]  
  
def get_name_labels(self):  
    return ", ".join([a.label for a in self.labels_available.all()])
```

Views.

new_accuracy.py: Se modificaron 2 funciones.

1. `CreateNewAccuracyView()`
2. `EditNewAccuracyView()`

la cuales reciben información para los 2 campos nuevos agregados en el modelo, esta información es opcional y los campos se llaman:

1. `category_labels`
2. `labels_category`

Los cuales se obtienen con `getlist()` ya que son una lista desde select multiples, pero se puede crear un un registro en el modelo sin esta información.

utils.py: Se modifiko los labels que muestra el reconocimiento según la selecciona de de labels y/o categoría de labels, recordando que esta nueva selección es opcional. También se agrego una función necesaria para la creación y edición de los registros en el modelo modificado.

Funciones que muestran labels según lo seleccionado al crear y/o editar, estas funciones son llamadas desde la view.

- `get_boxes_photo_task()`
- `get_boxes_photo_global()`

Dichas funciones también son necesarias validarse al momento de guardar la foto, la cual tiene una lógica al momento de ser llamada. Detalles que podrá verse en la sección de las views: **accuracy_photo_global.py** y **accuracy_photo_task.py** respectivamente.

Se agrego la función **GetLablesCategory()** que sirve para manejar la creación y/o edición de los registros en los nuevos campos del modelo, esta función es llamada desde las templates con Javascript, dicha funciona lo que hace es buscar basado en la selección de select dentro de dichas templates y construye un `<select></select>` HTML5 y lo devuelve como un Json.

accuracy_photo_global.py: Se modifiko la clase **AcurracyImageGlobalChangeView()** que muestra la información `get()` de las fotos en el accuracy de global, se modifica para mostrar la selección de labels o label category si tiene datos agregados.

```
"""
Obtener la categoria de los labels a mostrar
"""
ids_catory_labels = queryset.get_category_labels()
ids_labels = queryset.get_ids_labels()

boxes = get_boxes_photo_global(
    photo_global, country, ids_catory_labels, ids_labels
)
```

Como se puede ver en la foto, se toma la variable **queryset** que referencia el objeto del modelo modificado y llama los campos nuevos agregados mediante las funciones que se crearon dentro del modelo y se envían a la función **get_boxes_photo_global()**. De esta manera se aplica la lógica necesaria.

Si estos campos no tienen información se envía una lista vacía y para eso se aplica la lógica dentro de **get_boxes_photo_global()**.

También se modifica la clase **SaveAcurracyGlobalEditView()** de la misma manera que la clase anterior pero referenciando un `True` al final, también se puede validar en la lógica de la función.

```
"""
Obtener la categoria de los labels a mostrar
"""
ids_catory_labels = queryset.get_category_labels()
ids_labels = queryset.get_ids_labels()

boxes = get_boxes_photo_global(
    photo, request.country, ids_catory_labels, ids_labels, True
)
```

accuracy_photo_task.py: las clases modificadas en este archivo son:

1. AccuracyImagesChangeView()
2. SaveAccuracyEditView()

Las cuales son llamadas y referenciadas igual que la explicación de **accuracy_photo_global.py** la diferencia es que llama a su función específica, **get_boxes_photo_task()** pero aplicando la misma lógica anteriormente explicada.

Urls.

Se modificó el archivo **ebisuu_accuracy/urls/urls.py** para agregar la nueva función **GetLablesCategory()** en la **view ebisuu_accuracy/views/urls.py**.

Templates.

create_new_accuracy.html: Se agrega 2 select múltiples para el manejo de los 2 nuevos campos agregados al modelo.

```
<div class="col-lg-12">
  <label class="form-label font-bold">{% trans "Seleccionar Categoria de Labels
a Mostrar" %} *(Opcional)</label>
  <select id="select2_labels" class="js-example-basic-multiple"
multiple="multiple" name="category_labels">

  </select>
</div>

<div class="col-lg-12">
  <label class="form-label font-bold">{% trans "Seleccionar Labels" %}
*(Opcional)</label>
  <div id="group_label">

  </div>
</div>
```

El primer campo manejar los labelsCategory y utiliza la librería **select2.min.js**, la acción de dicho campo afecta el resultado del campo 2.

El segundo campo maneja los Labels pero es dependiente de la información seleccionada en el campo 1, este select maneja la librería **jquery.multi-select.js** y se construye su HTML en una función Javascript, dicha función javascript está en el mismo archivo de la template.

Dicha función javascript es la encargada de hacer una petición Ajax a la función nueva **GetLablesCategory()** contenida en el archivo **ebisuu_accuracy/views/urls.py**

```
// get Labels od ategory
function getLabelsCategory() {
    var id_country = $("#country_rule").val();
    var id_canal = $("#channels").val();
    var id_category_labels = $("#select2_labels").val();

    if (id_category_labels.length <= 0) {
        $("#group_label").html("");
    };

    if (id_country && id_canal && id_category_labels.length > 0) {
        var request = $.ajax({
            type: "GET",
            url: "{% url 'accuracy:utils:get_labels' %}",
            data: {
                "country": id_country,
                "canal": id_canal,
                "labels_category": id_category_labels
            },
        });
        request.done(function(response) {
            $("#group_label").html(`
                <select id="optgroup_labels" class="ms label_by_category" multiple="multiple"
                name="labels_category">
                    ${response.ResultLabel}
                </select>
            `);
            init_multi_select();
        });
    } else {
        $("#optgroup_labels").html("<option value='' selected='selected'>No Hay Resultados</option>");
    }
}
}
```

edit_new_accuracy.html: La información nueva de dicho archivo es la misma que en la template anterior, su diferencia va a a que modifica lo que existe y no agrega, también hay condicionales para mostrar la información agregada de Labes y labelCategory en caso se haberse agregado y la condición que puede modificarse los Labes y labelCategory si no se ah comenzado a revisar el paquete, de lo contrario esta opción no aparecerá. Tendrá el siguiente mensaje.

Ya se comenzó a revisar el paquete {{queryset.name_accuracy}}
 por ende no se puede cambiar las categorias/labels seleccionada.